IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Inventor(s): Alan Karp et al.

Confirmation No: 2634

Application No: 10/796690 **Examiner:** SAN JUAN, MartinJeriko P

Filing Date: March 8, 2004 **Group Art Unit:** 2109

Title: A System And Method For Safely Executing Downloaded Code On A Computer System

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

This Appeal Brief is presented in support of the Notice of Appeal filed on July 28, 2008, from the Final Rejection mailed April 28, 2008 rejecting claims 1-25 of the above-identified application.

Appellant respectfully requests reversal of the Examiner's rejection of pending claims 1-25.

CERTIFICATE OF TRANSMISSION
I hereby certify that this document is being transmitted to the Patent and Trademark Office via electronic filing on the date shown below.

October 28, 2008
Date of Transmission

Eileen Lehmann
Signature of Person Transmitting Papers

Eileen Lehmann
Typed or Printed Name of Person Transmitting Papers

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, LP having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware corporation, headquartered in Palo Alto, CA.

RELATED APPEALS AND INTERFERENCES

Appellant is not aware of any related appeals or interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal.

STATUS OF CLAIMS

Claims 1-25 are pending in the application (see Claims Appendix), and are the subject of the present Appeal.

Claims 1 and 3-9 were rejected under 35 U.S.C. 102(e) as being anticipated by Mackay et al. (US 7,107,448) hereafter "Mackay."

Claims 10-14 and 17-19 were rejected under 35 U.S.C. 102(e) as being anticipated by Charbonneau (US 2003/0074567) hereafter "Charbonneau."

Claims 2, 10, 15-16 and 20-25 were rejected under 35 U.S.C. 103(a) as being unpatentable over Mackay and further in view of Charbonneau.

STATUS OF AMENDMENTS

No amendments have been entered subsequent to the Final Rejection mailed April 28, 2008. The claims listed in the Claims Appendix, therefore, reflect the claims as of April 28, 2008.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Below the claims are described with reference to examples of support in the specification.

Claim 1

One aspect of the present invention, as claimed in independent claim 1, provides a a method (Figure 3; see pages 11, line 5 to page 12, line 16) for safely executing downloaded code on a computer system comprising accessing an application process wherein said

application process makes a system call to a library of said computer system for a resource, establishing a requesting thread. (Figure 3, 301, 303, p. 11, lines 5-16). The method further comprises sending a request message from said library to a local security filter (Figure 3, 305, p. 11, lines 18-23), validating said requesting thread at said local security filter and returning a digital signature that uniquely identifies said requesting thread to said application process (Figure 3, 307, p. 12, lines 1-7), and making a system call from said application process to a kernel of said computer system wherein said kernel uses said digital signature from said security filter to validate said requesting thread before allowing access to said resource (Figure 3, 309, p. 12, lines 9-16).

Claim 2

The method of Claim 1 can further comprise sharing a secret between said security filter and said kernel wherein said secret is used by said security filter to generate said digital signature and is used by said kernel to validate said digital signature. Figure 1 provides an illustrative example in which the kernel 130 is modified to share secret 125 with the security filter 120. By sharing the secret 125, the system kernel can verify the security filter signed the request. (Figure 1, p. 9, lines 17-20). Figure 4 and its discussion at p. 12, line 21 to p.14, line 19 also provides support for claim 2.

Claim 3

In one embodiment of the method as recited in claim 1, the library is a standard ntdll.dll library. (See p. 8, lines 9-10).

Claim 4

The method as recited in Claim 1 can further comprise restricting said security filter to an address space that is not accessible by said application. (See p. 11, lines 18-23).

Claim 5

The method as recited in Claim 1 can further comprise said kernel denying access to said resource if said digital signature can not be validated (See p. 14, lines 6-8).

Claim 6

The method as recited in Claim 1 can further comprise downloading executable code initiating said application process (See p. 12, lines 21-23, "As stated above, the resource request can be in response to code downloaded, e.g., from the Internet.")

Claim 7

The method as recited in Claim 1 can further comprise modifying said kernel such that only system calls that pass through said local library are allowed by said kernel. For example, in the discussion of the embodiment of Figure 1 at p. 9, lines 17-23, "the kernel 130 is also modified such that it too shares secret 125 with the security filter 120. By sharing the secret 125, the system kernel can verify that the security filter signed the request. If the request is properly signed, the resource request can be processed. If the request not is properly signed, it can be determined that the resource request did not pass through the local library 115 and the resource request is denied."

Claim 8

The method as recited in Claim 1 can further comprise restricting access of said application process to said resource for one command based on said digital signature. (See p. 12, lines 4-7.)

Claim 9

The method as recited in Claim 1 can further comprise restricting access of said application process to said resource for one time based on said digital signature. (See p. 12, lines 4-7.)

Claim 10

Claim 10 recites a method for determining the source of a resource request, and examples of support can be found in Figure 4 and its discussion at p. 12, line 21 to p.14, line 19 and Figure 2 and its discussion at pp. 10, line 13 to p. 11, line 3. The method comprises accessing a resource request associated with an application (401) p. 12, lines 21-23 and routing said resource request to a security filter, said security filter comprising a validation secret (403), p. 13, lines 1-8. Additionally, Figure 2 illustrates an example of a system for implementing the method where a resource request 145 is routed to a trusted security filter 120 which comprises a secret 125.

The method further comprises validating said resource request at said security filter (e.g. 120) and generating a first check value associated with said resource request using said validation secret (e.g. 125). Example support can be found in the discussion of Figure 4 (405) p. 13, lines 11-16.

Additionally, the method comprises routing said resource request to a system kernel (e.g. Resource Request transfer between Local Library 115 and OS Kernel 130 in Figure 2) wherein said system kernel comprises said validation secret (e.g. 125). Example support can be found in the discussion of Figure 4 (404) p. 13, lines 18-24.

The system kernel (e.g. 130) generates a second check value (e.g. Digital Signature for Resource Request between OS Kernel 130 and Digital Signature Hash 235) associated with said resource request based on said validation secret (see example support, Figure 4, 409, p. 13, lines 20-24) and allows access to said resource if said first check value and said second check value match. (see Figure 4, 410, p. 14, lines 1-8).

Claim 11

The method as recited in Claim 10 can further comprise denying access to said resource if said first check value and said second check value are different (See p. 14, lines 1-8).

Claim 12

The method as recited in Claim 10 can further comprise storing said first check value in a secure address space that is not accessible to said application. (See p. 11, lines 18-23).

Claim 13

The method as recited in Claim 10 can further comprise said system kernel retrieving said first check value from said secure address space. (See p. 13, lines 3-5 in the context of the discussion of Figure 4.)

Claim 14

The method as recited in Claim 10 can further comprise wherein said first check value is a digital signature. (See p. 13, line 12).

Claim 15

The method as recited in Claim 10 can further comprise restricting access of said application to said resource for a single resource request. (See p. 12, lines 4-7).

Claim 16

The method as recited in Claim 10 can further comprise restricting access of said application to said resource for a single time. . (See p. 12, lines 6-7).

Claim 17

The method as recited in Claim 10 can further comprise allowing only resource requests that pass through said security filter to be processed by said system kernel. (See 410 and the discussion at p. 14, lines 1-8).

Claim 18

The method as recited in Claim 10 can further comprise downloading executable content using said application (See p. 10, lines 1-4).

Claim 19

The method as recited in Claim 10 can further comprise modifying said kernel such that only system calls that pass through said security filter are processed by said kernel. (See discussion of modified kernel 130 at p. 9, line 17 to p. 10, line 7.)

Claim 20

Claim 20 recites a computer system for making it safe to execute downloaded code (For example, see discussion of Figure 1 at pages 8-10; Figure 2 and its discussion at pages 10. line 13 to p. 11, line 3) comprising

a modified local library (115) associated with an application (110), said local library coupled to a security filter (120) wherein said security filter comprises a secret (125) for generating a first digital signature (140, Digital Signature stored 269 in Figure 2) associated with a resource request from said application; and

a system kernel (130) for processing said resource request, said system kernel comprising said secret (125) for generating a second digital signature associated with said resource request wherein said kernel denies said resource request if said first digital signature and said second digital signature are different (See p. 10, lines 6-7, "If the two validation keys do not match, the request can be processed. If the two validation keys do not match, the request is denied." (For example, see discussion of Figure 1 at pages 8-10; Figure 2 and its discussion at pages 10. line 13 to p. 11, line 3).

Claim 21

The system as recited in Claim 20 wherein said application is a web browser (See p. 8, lines 6-8 and lines 15-19).

Claim 22

The system as recited in Claim 20 wherein said local library is a ntdll.dll library. (See p. 8, lines 9-10).

Claim 23

The system as recited in Claim 20 wherein said security filter is located in an address space that is not accessible by said application. (See p. 11, lines 18-23).

Claim 24

The system as recited in Claim 20 wherein said digital signature verifies that said resource request originated from said local library. (See p. 10, lines 4-11 in reference to Figure 1).

Claim 25

The system as recited in Claim 24 wherein said system kernel distinguishes between resource requests that come from said local library and resource calls that come from outside said local library wherein only resource calls that come from said local library are processed. As per the discussion for claim 7, please see, for example, the discussion of the embodiment of Figure 1 at p. 9, lines 17-23.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellant seeks review of the rejection of claims 1 and 3-9 under 35 U.S.C. 102(e) as being anticipated by Mackay et al. (US 7,107,448) hereafter "Mackay."

Appellant seeks review of the rejection of claims 10-14 and 17-19 under 35 U.S.C. 102(e) as being anticipated by Charbonneau (US 2003/0074567) hereafter "Charbonneau."

Appellant seeks review of the rejection of claims 2, 10, 15-16 and 20-25 under 35 U.S.C. 103(a) as being unpatentable over Mackay and further in view of Charbonneau.

ARGUMENT

Rejection of Claims 1 and 3-9 Under 35 U.S.C. § 102(e) as being anticipated by Mackay

Claims 1 and 3-9 were rejected under 35 U.S.C. 102(e) as being anticipated by Mackay et al. (US 7,107,448) hereafter "Mackay." Applicant respectfully asserts that claims 1 and 3-9 are patentable over Mackay.

Mackay is concerned with enabling content owners to supervise access to electronic content. "If it determined that the second application is not adequately enforcing the rules, the supervisory management system can revoke the second application's ability to access the content and/or the second application's ability to grant access to the content." (Col. 2, lines

31-35). The invention of claim 1 is directed to a method for safely executing downloaded code on a computer system. The method of claim 1 "validat[es] said requesting thread at said local security filter and return[s] a digital signature that uniquely identifies said requesting thread to said application process." The kernel of the computer system in claim 1 uses "said digital signature from said security filter to validate *said requesting thread* [emphasis added]."

Mackay is not validating threads, but the rights of applications to access rights managed content. (See col. 9, lines 8-12) "The mediator/shim 354 of the conformance library 351 may incorporate additional logic which, e.g. (a) allows it to verify or validate that a legitimate authorization certificate/ conformance certificate has been given by the content owner ..." An implementation example for the Governance engine of Mackay is "InterTrust's InterRights Point software or Rights/System software." Mackay is concerned with making sure an improperly modified application does not have rights to contents for digital rights management purposes, and not for confining threads for safely execution of downloaded code on a computer system.

Applicants respectfully assert that Mackay does not anticipate claims 1 and 3-9.

It is respectfully requested that the Board reverse this 35 U.S.C. § 102(e) Final Rejection.

Rejection of Claims 10-14 and 17-19 Under 35 U.S.C. § 102(e) as being anticipated by Charbonneau

Claims 10-14 and 17-19 were rejected under 35 U.S.C. 102(e) as being anticipated by Charbonneau (US 2003/0074567) hereafter "Charbonneau." Applicant respectfully asserts that claims 10-14 and 17-19 are patentable over Charbonneau. Charbonneau's paragraph [0035] fails to disclose the "method for determining the source of a resource request" as recited in independent claim 10. In its paragraph [0035] Charbonneau discloses that a password for a user is verified. "In use, a user of system 11 initiates an action requiring a password, such as for instance attempting to access a user data file 2 associated with the untrusted application 1." The system in Charbonneau does not generate "a first check value associated with said resource request." Instead, it compares the current state of the applications running within the computer system as represented by a hash value to a trusted

hash value retrieved when the system was in a verified secure state. These hash values are not associated with the resource request itself as are the first and second check values of claim 10.

Furthermore, Figure 2 of Charbonneau only indicates "a trusted group of applications" 4. Assuming arguendo that the trusted hash value is considered a "validation secret", Charbonneau fails to illustrate "a security filter comprising a validation secret" and "said system kernel comprising said validation secret." There is no indication these applications are in the kernel space, and likely may be in the application space. Furthermore, the trusted hash value does not appear to be comprised within a security filter or a system kernel.

Applicants respectfully assert that Charbonneau fails to anticipate independent claim 10, and hence its dependent claims 11-19. It is respectfully requested that the Board reverse this 35 U.S.C. § 102(e) Final Rejection.

Rejection of Claims 2, 10, 15-16 and 20-25 Under 35 U.S.C. § 103(a) as being unpatentable over Mackay and further in view of Charbonneau

Claims 2, 10, 15-16 and 20-25 were rejected under 35 U.S.C. 103(a) as being unpatentable over Mackay and further in view of Charbonneau. Applicant respectfully asserts that claims 2, 10, 15-16 and 20-25 are patentable over Mackay in view of Charbonneau.

The arguments presented above for independent claims 1 and 10 are applicable for illustrating why claim 2 which depends from claim 1 and claim 10 and its dependents 15-16 are patentable over the combination of Mackay in view of Charbonneau.

Furthermore, the combination of Mackay in view of Charbonneau fails to render unpatentable claims 20-25. Neither of these references teaches "a security filter [which] comprises a secret for generating a first digital signature" as well as a "system kernel comprising said secret for generating a second digital signature associated with said resource request."

CONCLUSION

It is respectfully requested that the Board reverse the final rejections of claims 1-25 under both 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a).

Respectfully submitted,

Alan Karp et al.

Eileen Lehmann

Oct. 28, 2008

Eileen Lehmann
Registration No. 39,272
Hewlett-Packard Company
Mail Stop 1197
1501 Page Mill Road
Palo Alto, CA 94304
650-857-7940 (telephone)
650-852-8063 (fax)

CLAIMS APPENDIX

1. (Original) A method for safely executing downloaded code on a computer system comprising:

accessing an application process wherein said application process makes a system call to a library of said computer system for a resource, establishing a requesting thread;

sending a request message from said library to a local security filter;

validating said requesting thread at said local security filter and returning a digital signature that uniquely identifies said requesting thread to said application process; and

making a system call from said application process to a kernel of said computer system wherein said kernel uses said digital signature from said security filter to validate said requesting thread before allowing access to said resource.

2. (Original) The method as recited in Claim 1 further comprising

sharing a secret between said security filter and said kernel wherein said secret is used by said security filter to generate said digital signature and is used by said kernel to validate said digital signature.

3. (Original) The method as recited in Claim 1 wherein said library is a standard ntdll.dll library.

4. (Original) The method as recited in Claim 1 further comprising:

restricting said security filter to an address space that is not accessible by said application.

5. (Original) The method as recited in Claim 1 further comprising:

said kernel denying access to said resource if said digital signature can not be validated.

6. (Original) The method as recited in Claim 1 further comprising:

downloading executable code initiating said application process.

7. (Original) The method as recited in Claim 1 further comprising:

modifying said kernel such that only system calls that pass through said local library are allowed by said kernel.

8. (Original) The method as recited in Claim 1 further comprising:

restricting access of said application process to said resource for one command based on said digital signature.

9. (Original) The method as recited in Claim 8 further comprising:

restricting access of said application process to said resource for one time based on said digital signature.

10. (Original) A method for determining the source of a resource request comprising:

accessing a resource request associated with an application;

routing said resource request to a security filter, said security filter comprising a validation secret;

validating said resource request at said security filter and generating a first check value associated with said resource request using said validation secret;

routing said resource request to a system kernel wherein said system kernel comprises said validation secret;

generating a second check value associated with said resource request based on said validation secret at said system kernel; and

allowing access to said resource if said first check value and said second check value match.

11. (Original) The method as recited in Claim 10 further comprising:

denying access to said resource if said first check value and said second check value are different.

12. (Original) The method as recited in Claim 10 further comprising:

storing said first check value in a secure address space that is not accessible to said application.

13. (Original) The method as recited in Claim 12 further comprising:

said system kernel retrieving said first check value from said secure address space.

14. (Original) The method as recited in Claim 10 wherein said first check value is a digital signature.

15. (Original) The method as recited in Claim 10 further comprising:

restricting access of said application to said resource for a single resource request.

16. (Original) The method as recited in Claim 10 further comprising:

restricting access of said application to said resource for a single time.

17. (Original) The method as recited in Claim 10 further comprising:

allowing only resource requests that pass through said security filter to be processed by said system kernel.

18. (Original) The method as recited in Claim 10 further comprising:

downloading executable content using said application.

19. (Original) The method as recited in Claim 10 further comprising:

modifying said kernel such that only system calls that pass through said security filter are processed by said kernel.

20. (Original) A computer system for making it safe to execute downloaded code comprising:

a modified local library associated with an application, said local library coupled to a security filter wherein said security filter comprises a secret for generating a first digital signature associated with a resource request from said application; and

a system kernel for processing said resource request, said system kernel comprising said secret for generating a second digital signature associated with said resource request wherein said kernel denies said resource request if said first digital signature and said second digital signature are different.

21. (Original) The system as recited in Claim 20 wherein said application is a web browser.

22. (Original) The system as recited in Claim 20 wherein said local library is a ntdll.dll library.

23. (Original) The system as recited in Claim 20 wherein said security filter is located in an address space that is not accessible by said application.

24. (Original) The system as recited in Claim 20 wherein said digital signature verifies that said resource request originated from said local library.

25. (Original) The system as recited in Claim 24 wherein said system kernel distinguishes between resource requests that come from said local library and resource calls that come from outside said local library wherein only resource calls that come from said local library are processed.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None